网络应用开发课程设计

基于X3DOM框架的Web3D操纵网站

**组员信息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 姓名 | 学号 | 成员贡献 | 课程设计成绩 |
| 1 | 王晓雨 | 2022280075 |  |  |
| 2 | 李易婷 | 2022280079 |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

Github代码仓库URL：https://github.com/wxy2000/wxy1.git

# **引言**

JavaScript是创建Web网站最有用的编程语言之一，尤其是在WebGL库的支持下。WebGL是JavaScript API或基于OpenGL的库，它使Web能在浏览器中执行三维和二维图形，而无需其他插件。WebGL渲染到计算机的图形处理单元，以超快的速度呈现复杂的概念、动画和科学模拟。有了WebGL，可以很方便地使用HTML5 Canvas元素动态生成图形。因此，设计师和开发人员很容易创建流畅的2D和3D效果。

而其中X3DOM是最流行的WebGL框架。它用于生成可嵌入的3D Web图形。适用于网站和应用程序。运行时性能以声明方式，没有任何低级JavaScript或GLSL编码。

本开发项目的目标是基于X3DOM框架构建一个Web网站，实现Web3D建模，采用标准的浏览器技术，在HTML DOM中实现一个实时的X3D场景，使用结构化的文本表示形式来创建和显示交互式3D场景，使用户仅添加，删除或更改DOM元素就可以操纵3D内容，不需要特定的插件或插件接口，使3D内容就像文本、链接、图像或电影一样，成为网页内容的一部分。输入该3D物体的图形参数，颜色，经过3D建模，且通过JavaScript编程，可以实现利用鼠标对该模型进行的操纵，包括对该3D模型进行拖拽，旋转，缩放。本项目是一项独立的模块，而且全部内容自含，应用场景可用于网络购物平台商品的3D展示，从而给消费者带来更直观形象的购物体验。

# 功能与设计

## 功能介绍

本项目实现的核心功能包括：

无需插件即可显示X3DOM场景，基于标准的浏览器技术，例如HTML5和WebGL；

输入该3D物体的图形参数，颜色，经过3D建模，且通过JavaScript编程，可以实现利用鼠标对该模型进行的操纵，包括对该3D模型进行拖拽，旋转，缩放；

开发DOM操作引擎库，取名为wqMV.js，存放在wqMV文件夹下，其功能类似于jQueryAPI，主要是HTML文档的DOM遍历和操作，以及事件处理；

自己创建wqMV库对js编程进行了简化操作；

系统应支持的终端数取决于Web服务器所支持的最大终端数，一般为几百万台，应支持的并行操作的用户数为Web服务器所设定的并发用户数量，一般为几十万个；

应用场景可用于网络购物平台商品的3D展示，从而给消费者带来更直观形象的购物体验。

## 系统设计

### 2.2.1关键设计

本项目关键设计部分包括index.html主界面，index.js主控模块，wqMV库，X3DOM框架，css样式表。

|  |  |  |
| --- | --- | --- |
| 模块名称 | 功能 | 接口 |
| index.html主界面 | 呈现整个网页 | 无 |
| index.js主控模块 | 调用wqMV库和x3DOM框架，修改html文件 | 无 |
| wqMV库 | HTML 文档的 DOM 遍历和操作，以及事件处理，简化js的编写 | q，m |
| x3DOM框架 | 实现3d模型的呈现，缩放，旋转，渲染，实物模拟 | x3d |
| css样式表 | 设计网页样式 | 无 |

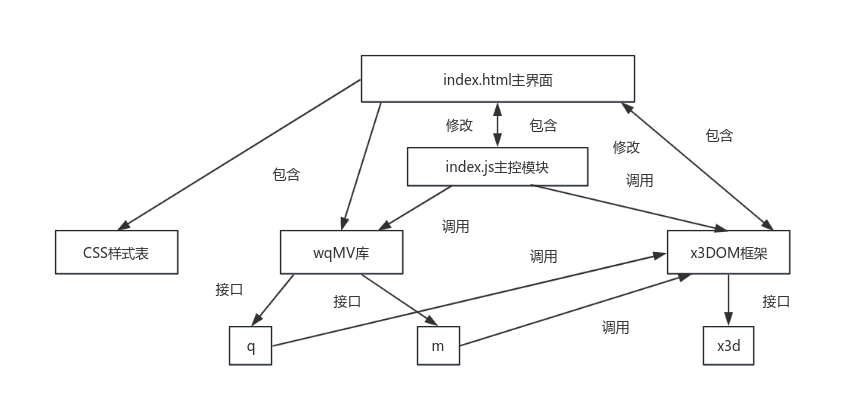
### 2.2.2系统架构

本网站整体架构如下所示：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Hello Web3D! | | | | 显示日期 | | | |
| 基础开发 | | | | | | | |
| 展示圆锥，正方形，圆球 | | | | | | | |
| 动画 | | | | | | | |
| 展示红色小球上下弹跳 | | | | | | | |
| 高级渲染 | | | | | | | |
| 展示图片，视频，声音渲染 | | | | | | | |
| 实物模拟 | | | | | | | |
| 展示小鹿模型 | | | | | | | |
|  | 点我 | 前面 | 右面 | 最佳视野 | 定向光 | 射灯 |  |

### 2.2.3逻辑流程

本项目的整体流程图如下所示：



其中定义了一些系统或产品中涉及的重要术语，为读者在阅读文档时提供必要的参考信息：

|  |  |  |
| --- | --- | --- |
| 序号 | 术语或缩略语 | 说明性定义 |
| 1 | q | Quick，基于DOM快速操作及事件处理 |
| 2 | m | Model，基于文档的排版及校验、日期等操作 |
| 3 | x3d | 引用X3DOM框架 |

### 2.2.4接口设计

1．wqMV库中q接口：Quick，基于DOM操作及事件处理。

q('idname')获取id为idname的网页元素节点；

q('idname').h获取或者设置id为idname网页元素的innerHTML；

q('.class')获取className为class的所有节点；

q('@LI')获取元素标签为li的所有节点；

2.wqMV库中m接口：Model，基于文档的排版及校验、日期等操作。

m.ut()获取当前日期；

3.x3d接口: 引用X3DOM框架

X3D基于XML格式开发，所以可以直接使用XMLDOM文档树、XMLSchema校验等技术和相关的XML编辑工具。目前X3D已经是通过ISO认证的国际标准。

可使用Maya、Blender、3DsMax或者WorldofWarcraft等软件来建立X3D类型的3D模型文件。

网站添加x3d格式MIME类型。加载x3d模型文件。

### 2.2.5系统说明

1.系统运行环境：简单服务器，Windows操作系统，VScode编程平台。

2.开发环境：常见的Web服务器，如apache和IIS等。

3.通信协议：http网络协议。

4.数据类型：

（1）常量：

常量对象wxy：所在文件为index.js文件，属于全局常量，功能为主控程序。

常量对象q：所在文件为wqMV.js文件，属于全局常量，功能为基于 DOM 操作及事件处理。

常量对象m：所在文件为wqMV.js文件，属于全局常量，功能为基于文档的排版及校验、日期等操作。

（2）变量：

wqMV.css文件中的root为变量。

deer.x3d文件为变量。

鼠标的点击，拖拽，滚轮输入为变量，3d模型的颜色，大小，方向的为变量。

日期的获取为变量。

# **功能实现**

## 3.1关键技术

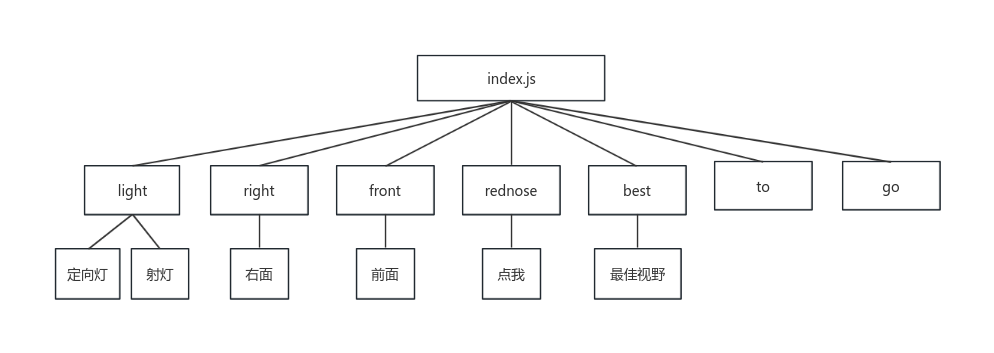
### 3.1.1 index.js主控模块

调用wqMV库和x3DOM框架，修改html文件，实现“点我”，“前面”，“右面”，“最佳视野”，“定向光”，“射灯”按钮的功能，通过鼠标左键点击产生3D模型颜色变化，大小变化和方向变化。

数据常量为一个名为wxy的对象。数据存储在index.js文件中。

首先创建一个wxy的常量对象，然后在该对象中定义了“light”，“best”，“right”，“front”，“rednose”，“to”六个元素，每个元素为一个函数。其中，当鼠标点击对应按钮时，对应的函数会调用q接口获得相应数据进行操作，从而改变实物模拟的deer模型的方向，颜色，大小。

其设计图如图所示：



### 3.1.2 wqMV库

其功能类似于jQuery API，主要是HTML文档的DOM遍历和操作，以及事件处理。其中q接口：quick，基于DOM操作及事件处理；m接口：Model，基于文档的排版及校验、日期等操作。

输入数据为程序中用到的节点id以及属性名称，字符串形式，输出数据为节点的属性的变化。数据存储为wqMV.js文件中。

其中q子模块为一个匿名自执行函数，其中包括q函数和对push还有join指令的简化操作。其中q函数实现了四个方法分别为

g:设置或读取属性值；

t:获取文本数据；

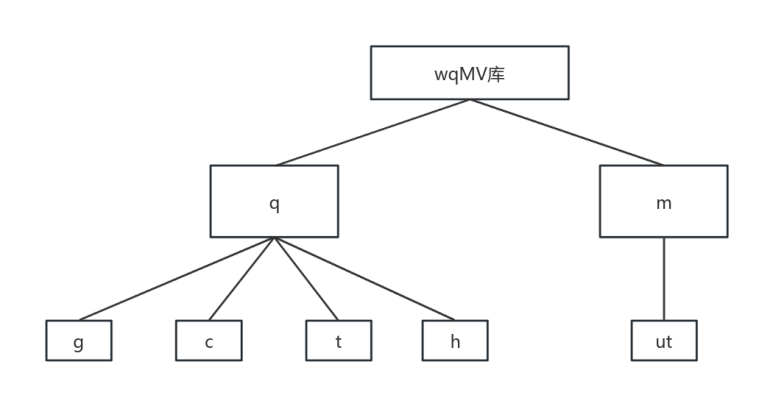
c:设置或读取classname；

h:设置或读取html数据。

并且对q函数的三个参数进行了矫正，另外运用判断语句对节点进行筛选，还实现了节点数组到数组的转换以方便对数组进行for循环操作。

m子模块也是一个匿名自执行函数，其中包括m常量对象，它的ut函数元素实现了年月日时分秒的获取，并对显示位数及中文汉字的显示进行了修正。

其设计图如图所示：



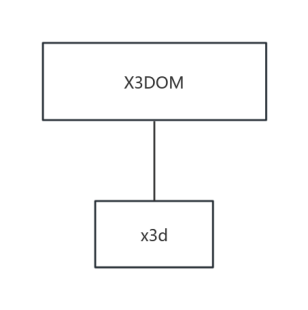
### 3.1.3 x3DOM框架

实现3d模型的呈现，缩放，旋转，渲染，实物模拟。

输入数据为鼠标输入设备或者x3d文件。有效性验证：x3d文件需符合XML标准。输出数据为3D模型的大小，方向，颜色的变化。

数据文件为js和css文件。保存的文件名为x3dom.css和x3dom.js。

其设计图如图所示：



### 3.1.4 css样式表

用于设计网页样式。

Root变量：

--ct: rgba(0, 0, 0, .8); /\* Color Text \*/

--cu: #08e; /\* Color blUe \*/

--gw: #fff; /\* backGround White \*/

--gh: azure; /\* backGround Hover \*/

--bdc: rgba(99, 99, 99, .382); /\* BorDer color \*/

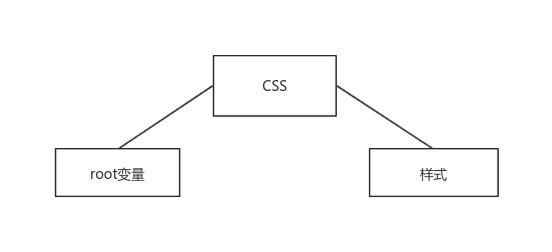
--bds: 1px solid var(--bdc); /\* BorDer Solid \*/

--bdd: 1px dotted var(--bdc); /\* BorDer Dotted \*/

--br: 3px /\* Border Radius \*/

输出网页的背景颜色，样式，按钮的样式以及划过时的变化。

其设计图如图所示：



### 3.1.5 index.html主界面

用于呈现整个网页。

输入数据为通过鼠标获取的数据，表现形式为滚轮，拖拽，以及html语言设计的网页文本数据。

输出浏览器网页形式。

# **功能展示与评测**

系统可用性和完整性

# **总结**

总结与问题